도면의 간단한 설명

제1도는 본 발명의 구성을 사용한 컴파일러의 개략도이다. 제2도는 본 발명의 다양한 특징을 갖는 방법이 실행될 수 있는 호스트 컴퓨터의 블록도이다. 제3a, 3b, 3c도는 제1도의 컴파일러에 의해 번역될 코드를 소스 코드 형태, 중간 언어 형태, 트리 형태(tree form) 및 어셈블리 언어 형태로 나타낸 도면이다.

FIG. 1

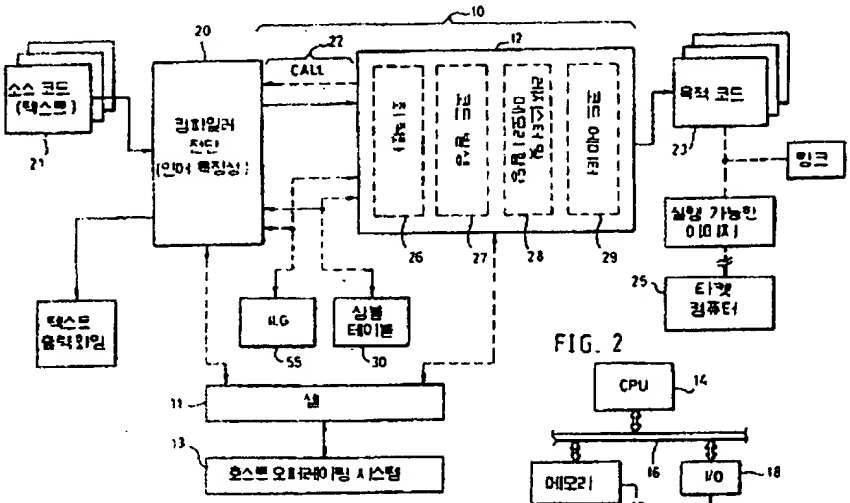

FIG. 2

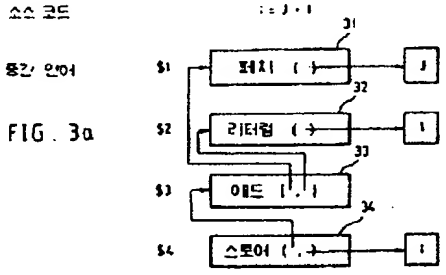

FIG. 3a

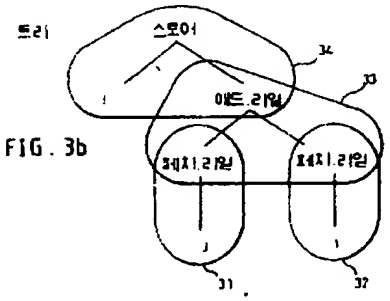

FIG. 3b

FIG. 3c

어셈블리

```
LOAD  4,J
ADDI  4,1
STORE 4,I
```

RISC

OR

```
ADD   #1,J,I
```

CISC

⑲대한민국특허청(KR)

제 1004 호

⑤Int. Cl.⁵
G 06 F 9/46

⑫국제특허출원의 출원공개공보 (A)

⑪공개번호 93-700916

⑬공개일자 서기 1993. 3. 16

⑳출원번호 92-702693

번역문제출일자 서기 1992. 10. 27

심사청구 : 있음

⑧국제출원번호 PCT/US 92/01278

⑧국제출원일자 1992. 2. 18

⑨지정국 : 오스트리아, 벨기에, 스위스&리히텐스타인, 독일, 덴마크, 스페인, 프랑스, 영국, 그리이스, 이태리, 룩셈부르크, 모나코, 네덜란드, 스웨덴, 베닌, 브르키나파소, 카메룬, 중앙아프리카공화국, 차드, 콩고, 코트디부와르, 가봉, 기니아, 말리, 마리타니아, 세네갈, 토고, 오스트리아, 호주, 바바도스, 불가리아, 브라질, 캐나다, 스위스&리히텐스타인, 체코슬로바키아, 독일, 덴마크, 스페인, 핀란드, 영국, 헝가리, 일본, 북한, 한국, 스리랑카, 룩셈부르크, 마다가스카르, 몽골리아, 말라위, 네덜란드, 노르웨이, 폴란드, 루마니아, 수단, 스웨덴, 러시아.

⑧국제공개번호 WO 92/15942

⑧국제공개일자 1992. 9. 17

⑩우선권주장 ⑩1991. 2. 27 ⑩미국(US) ⑩662,461
1991. 2. 27 662,725
1991. 2. 27 662,477
1991. 2. 27 662,483
1991. 2. 27 662,464

⑫발 명 자 플리크스테인, 데이비드, 스콧트
미합중국, 뉴햄프셔 03051, 허드슨, 로빈슨로드 96
데이비드슨, 캐롤라인, 스위니
미합중국, 뉴햄프셔 03049, 홀리스, 라이드아웃로드 155
페이만, 로버트, 네일, 쥬니어
미합중국, 뉴햄프셔 03806, 윕튼, 퍼트남 힐 로드
그로브, 리차드, 매티
미합중국, 매사츄세츠 01886, 웨스트포드, 캐리지 웨이 5
호브스, 스티븐, 오.
미합중국, 매사츄세츠 01886, 웨스트포드, 버터넛트로드 10
머피, 데니스, 조셉
미합중국, 매사츄세츠 01886, 웨스트포드, 데포트 로드 86

⑪출 원 인 디지탈 이큅먼트 코오포레이숀 대표자 로날드 이. 마이릭
미합중국, 매사츄세츠 01754, 메이나드, 메인 스트리트 146

⑭대리인 변리사 나 영 환·도 두 형

(전 3 면)

⑤다중 언어 최적화 컴파일러의 분석 유도 표현

⑤특허청구의 범위

1. 각종 다른 고급 프로그래밍 언어중 하나로 된 소스 코드를 포함하는 입력 코드 모듈에 정의된 프로그램을 언어 특정성을 갖는 컴파일러 전단을 사용하여 액세스하는 단계와, 상기 컴파일러 전단내의 변환기에 의해 중간 언어로 상기 프로그램을 정의하도록 중간 언어 그래프 및 심볼 데이블을 발생하는 단계와, 상기 중간 언어 그래프를 액세스하여 상기 중간 언어 그래프를 제구성함으로써 상기 프로그램을 최적화하는 단계와, 각종 다른 타겟 컴퓨터 아키텍처중 하나에 대한 코드 발생기에 의해 상기 중간 언어 그래프로부터 목적 코드를 발생

하는 단계를 포함하는데, 상기 최적화 단계가 상기 프로그램내의 유도 변수의 표현을 검출하고 그 유도 변수의 표현을 변경된 중간 언어 그래프에 의해 정의된 동등한 표현으로 변환하여, 승산을 가산으로 대체함으로써 상기 프로그램용의 상기 튜플로부터 생성된 상기 목적 코드의 실행 시간을 감소시키는 과정을 포함하는 것을 특징으로 하는 코드 번역 방법.

2. 제1항에 있어서, 상기 중간 언어 그래프가 상기 입력 코드 모듈에서의 단일 표현을 표시하는 튜플들로 구성되는 것을 특징으로 하는 코드 번역 방법.

3. 제2항에 있어서, 상기 중간 언어 그래프가 상기 튜플들의 순서를 포함하는 블록들로 구성되는 것을 특징으로 하는 코드 번역 방법.

4. 제3항에 있어서, 상기 각 블록들이 엔트리 라벨로 시작되어 중간 출구 없이 복귀로 끝나는 것을 특징으로 하는 코드 번역 방법.

5. 제3항에 있어서, 상기 튜플들이 연산자, 연산자 데이타 타입 및 피연산자를 표시하는 필드를 갖는 데이타 구조인 것을 특징으로 하는 코드 번역 방법.

6. 각종 다른 고급 프로그래밍 언어중 하나로된 소스 코드를 포함하는 입력 코드 모듈에 정의된 프로그램을 액세스하기 위한 언어 특정성을 갖는 컴파일러 전단과, 중간 언어 흐름 그래프를 발생하여 상기 프로그램을 중간 언어로 정의하기 위한 상기 컴파일러 전단내의 변환기와, 상기 중간 언어 그래프를 액세스하여 상기 중간 언어 그래프를 재구성함으로써 상기 프로그램을 최적화시키기 위한 최적화 수단과, 상기 중간 언어 그래프를 액세스함으로써 각종 다른 타겟 컴퓨터 아키텍처중 하나에 목적 코드를 발생하기 위한 코드 발생기를 구비하는데, 상기 최적화 수단이 상기 프로그램의 유도 변수의 표현을 검출하고 그 유도 변수의 표현을 변경된 중간 언어 그래프에 의해 정의된 동등한 표현으로 변환하여, 승산에 가산으로 대체함으로써 상기 프로그램용의 상기 튜플로부터 생성된 상기 목적 코드의 실행 시간을 감소시키기 위한 수단을 구비하는 것을 특징으로 하는 코드 번역 장치.

7. 제6항에 있어서, 상기 중간 언어 그래프가 각각 상기 입력 코드 모듈에서의 단일 표현을 표시하는 튜플들로 구성되는 것을 특징으로 하는 코드 번역 장치.

8. 제7항에 있어서, 상기 튜플들이 연산자, 연산자 데이타 타입 및 피연산자를 표시하는 필드를 갖는 데이타 구조인 것을 특징으로 하는 코드 번역 장치.

9. 제8항에 있어서, 상기 중간 언어 그래프가 상기 튜플들의 순서를 포함하는 블록들로 구성되는데 상기 각 블록들은 입구 또는 라벨로 시작하여 중간 출구없이 복귀로 끝나는 것을 특징으로 하는 코드 번역 장치.

10. 제9항에 있어서, 상기 중간 언어 그래프를 액세스함으로써 상기 코드를 최적화하기 위한 상기 수단이 상기 튜플들의 결과 또는 종속성 표시에 응답하는 것을 특징으로 하는 코드 번역 장치.

11. 각각 소스 코드 언어의 단일 표현을 표시하는 튜플들로 구성되는 중간 언어 그래프를 언어 특정성을 갖는 컴파일러 전단을 사용하여 중간 언어로 발생하는 단계와, 상기 중간 언어 그래프의 유도 변수의 표현을 검출하고 그 유도 변수의 표현을 변경된 중간 언어 그래프에 의해 정의된 동등한 표현으로 변환하여, 승산을 가산으로 대체함으로써 상기 프로그램용의 상기 튜플로부터 생성된 목적 코드의 실행 시간을 감소시키는 단계를 포함하는 것을 특징으로 하는 컴파일러내의 코드 최적화 방법.

12. 제11항에 있어서, 상기 중간 언어 그래프가 상기 튜플들로 순서를 포함하는 블록들로 구성되는 것을 특징으로 하는 컴파일러내의 코드 최적화 방법.

※ 참고사항 : 최초출원 내용에 의하여 공개하는 것임.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :		(11) International Publication Number: WO 92/15942
G06F 9/45	A1	(43) International Publication Date: 17 September 1992 (17.09.92)

(21) International Application Number: PCT/US92/01278

(22) International Filing Date: 18 February 1992 (18.02.92)

(30) Priority data:

662,461	27 February 1991 (27.02.91)	US
662,725	27 February 1991 (27.02.91)	US
662,477	27 February 1991 (27.02.91)	US
662,483	27 February 1991 (27.02.91)	US
662,464	27 February 1991 (27.02.91)	US

(71) Applicant: DIGITAL EQUIPMENT CORPORATION [US/US]; 146 Main Street, Maynard, MA 01754 (US).

(72) Inventors: BLICKSEIN, David, Scott ; 96 Robinson Road, Hudson, NH 03051 (US). DAVIDSON, Caroline, Sweeney ; 155 Rideout Road, Hollis, NH 03049 (US). FAIMAN, Robert, Neil, Jr. ; Putnam Hill Road, Wilton, NH 03806 (US). GROVE, Richard, Barry ; 5 Carriage Way, Westford, MA 01886 (US). HOBBS, Steven, O. ; 10 Butternut Road, Westford, MA 01886 (US). MURPHY, Dennis, Joseph ; 86 Depot Road, Westford, MA 01886 (US).

(74) Agents: NATH, Rama, B. et al.; Joyce Lange (MS02-3/G3), Digital Equipment Corporation, 111 Powdermill Road, Maynard, MA 01754 (US).

(81) Designated States: AT, AT (European patent), AU, BB, BE (European patent), BF (OAPI patent), BG, BJ (OAPI patent), BR, CA, CF (OAPI patent), CG (OAPI patent), CH, CH (European patent), CI (OAPI patent), CM (OAPI patent), CS, DE, DE (European patent), DK, DK (European patent), ES, ES (European patent), FI, FR (European patent), GA (OAPI patent), GB, GB (European patent), GN (OAPI patent), GR (European patent), HU, IT (European patent), JP, KP, KR, LK, LU, LU (European patent), MC (European patent), MG, ML (OAPI patent), MN, MR (OAPI patent), MW, NL, NL (European patent), NO, PL, RO, RU, SD, SE, SE (European patent), SN (OAPI patent), TD (OAPI patent), TG (OAPI patent).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: ANALYZING INDUCTIVE EXPRESSIONS IN A MULTILANGUAGE OPTIMIZING COMPILER

(57) Abstract

A compiler framework uses a generic "shell" or control and sequencing mechanism, and a generic back end (where the code generator is target-specific). The generic back end includes the functions of optimization, register and memory allocation, and code generation. The shell may be executed on various host computers, and the code generation function of the back end may be targeted for any of a number of computer architectures. A front end is tailored for each different source language, such as Cobol, Fortran, Pascal, C, C++, Ada, etc. The front end scans and parses the source code modules, and generates from them an intermediate language ("IL") representation of the programs expressed in the source code. This IL is constructed to represent any of the source code languages in a universal manner, so the interface between the front end and back end is of a standard format, and need not be rewritten for each language-specific front end. The IL representation generated by the front end is based upon a tuple as the elemental unit, where each tuple represents a single operation to be performed, such as a load, a store, an add, a label, a branch, etc. A data structure is created by the front end for each tuple, with fields for various necessary information. One feature of the invention is a mechanism for representing effects and dependencies in the interface between front end and back end; a tuple has an effect if it writes to memory, and has a dependency if it reads from a location which some other node may write to. A mechanism independent of source language is provided for describing the effects of program execution. Another feature is the use in the optimization part of the compiler of a method for analyzing induction variables, whre the improvement is to use the side effects sets used to construct IDEF sets. Another feature is a mechanism for "folding constants" (referred to as K-folding or a KFOLD routine), included as one of the optimizations. A further feature is the type definition mechanism, referred to as the TD module, which provides mechanisms used by the front end and the compiler of the back end in constructing program type information to be incorporated in an object module for use by a linker or debugger. Another feature is a method for doing code generation using code templates in a multipass manner.